24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

From mixture of longitudinal and non-Gaussian advertising data to Click-Through-Rate prediction

Faustine Bousquet [1,2] and **Sophie Lebre** [3,1] and **Christian Lavergne** [3,1]

Abstract. Click-Through-Rate (CTR) prediction is one of the most important challenges in the advertisement field. Nevertheless, it is essential to understand beforehand the voluminous and heterogeneous data structure. Here we introduce a novel CTR prediction method using a mixture of generalized linear models (GLMs). First, we develop a model-based clustering method dedicated to publicity campaign time-series, i.e. non-Gaussian longitudinal data. Secondely, we consider two CTR predictive models derived from the inferred clustering. The clustering step improves the CTR prediction performance both on simulated and real data. An R package *binomialMix* for mixture of binomial and longitudinal data is available on CRAN.

1 Introduction

1.1 Context

The field of advertising, and more particularly online advertising, has been disrupted by the development and success of Real-Time Bidding (RTB) [20, 21]. This process connects advertisers and publishers in real time and gives them the advantage of personalization via an auction system: the publisher provides a set of information about the ad slot context and the advertiser can decide whether he is interested in the auction or not. This system reduces ineffective targeting. We call "impression" the advert's display on the end user's device. The Click-Through Rate (CTR) is the most common way to estimate the efficiency of an advertising campaign. It measures the ratio of the number of times the user has clicked the ad and the number of times the ad has been displayed. Many statistical challenges emerged from online advertising including CTR prediction [2, 3, 9, 19].
In this paper, we address a real-world online issue from TabMo[2] advertising platform. TabMo is an adtech company managing the campaigns for the advertisers. Its business objective is to provide the best ad slot context for every campaign and increase Key Point of Interest as CTR for advertisers. Constraints coming from production context make the issue interesting in many ways. The data volume is huge. Every second the predictive model has to answer around 1 million auctions with the most adequate advertising campaign among all available. Also, we are in case of rare events: number of click is very low compared to the number of impressions (around 1 click every 1000 impressions). The very imbalanced data makes predictions difficult. Despite all the studies (see Section 1.2) conducted on the prediction of user response in an online advertising context, the prediction of CTR remains an open and still relevant issue.

[1] IMAG Institut Montpellierain Alexander Grothendieck, Univ. Montpellier, CNRS, France
[2] TabMo Labs, Montpellier, France - https://hawk.tabmo.io/
[3] Universit Paul Valry Montpellier 3, France

1.2 Related work

The last three years, Neural Network emerged in the online advertising state-of-art for CTR prediction. At the same time, the dimension of features and the data volume has increase. That is one of the reasons why a lot of research and continuous improvements have been done in model structure Neural Network. [3] developed an hybrid predictive model using both the advantages of linear model and deep network architecture. Predictions from both component are combined to produce a final prediction. In [19], the proposed architecture of the DNN focuses on taking into account interactions between variables beyond order 2. Without the need of a manualy preprocessing data and with a quite simple implementation for this kind of modeling, authors assume that there is a significant decrease of the logloss value compared to a classical DNN architecture. Neural networks take advantage of their multi-layered architecture and achieve good predictive results. However, their complexity makes it difficult to understand the model.
Factorization Machines (FM) [16] models second order polynomial with a latent vector for each feature. The interaction between two features is calculated by the inner product of the latent vector from those variables. The advantage of this method is that it reduces the complexity of the model when interactions are taken into account. A lot of extensions emerged from FM [6, 8] in the context of CTR prediction where capturing the information of feature conjunctions can be relevant. [8] proposed a modeling based on levels features interactions while [6] combined the pertinence of Neural Network with FM in the same architecture. However, until now, the Factorization Machines are mostly used for recommender systems.
Predicting CTR using logistic regression is also one of the most studied models in the literature [2, 9]. Logistic regression models present the advantage of an easy implementation for real-world predictive problem. However, to model more complex and heterogeneous data structures from real case studies, the use of logistic regression may be limited. The use of mixture models allows for better consideration of heterogeneity and data specificity.

Mixture model of Generalized Linear Model (GLM) is a well-studied statistical problem in the literature; [12] gives a complete overview of different existing methodologies for model-based clustering. The number of R packages have grown significantly in various domains of application (such as biology, physics, economics...) to model data with a finite number of subpopulations. *Mixmod* [18] is a popular tool for partitioning observations into groups. The package allows to fit Gaussian mixture models with different volume, shape and orientation of the clusters. It can also model mixtures of multinomial distribution. The *mclust* [5] package is another tool for finite

normal mixture modeling. In the case of longitudinal data, the package *flexmix* [10] provides a mixture modeling when there are M individuals with respectively n_M observations. But to the best of our knowledge, the existing packages cannot model longitudinal data in a binomial context

1.3 Contribution

This paper addresses a real-world online advertising issue. Using the traffic log of a set of campaigns, we build an efficient method that predicts a context specific click probability for each campaign. Our main contributions are the following:

- We describe a GLM-based clustering approach for longitudinal data in a binomial context. Campaigns are clustered according to their CTR depending on the advertising context described by continuous and categorical variables. The longitudinal data is defined as repeated observations for each campaign with a specific length. The temporal periodicity is captured via 2 categorical variables: day and time slot. Time slot separation was established with domain experts. This model-based clustering allows us to group advertising campaigns with similar profiles.
- We predict context specific CTR for each campaign using this model-based clustering as a preliminary step. Various clustered-based prediction schemes are considered.
- Using GLM mixture for CTR prediction leads to good performance while preserving a simple model architecture and a rapid deployment process. Experiments are performed on data extracted from a real-world online advertising context.
- We developed an R package: *binomialMix* (available on CRAN) implementing a GLM-based clustering approach for longitudinal data in a binomial context.

1.4 Outline

Section 2 presents the two-step statistical modeling and the resulting implementation. The first step describes the mixture of binomial for longitudinal data estimated with an Expectation-Maximization (EM) algorithm. The second step builds a predictive model to provide a probability of click using the clustering step. In Section 3, we (i) present the dataset, the evaluation metrics, the results on simulated data and (ii) challenge five predictive models on real data. Three are considered as baseline. The two others use mixture model estimations to predict a probability of click in a given context. We evaluate the relevance of clustering and the performance of the predictions.

2 Proposed Approach

2.1 A binomial model for the CTR

The proposed approach to address the problem of CTR prediction is described in two steps. First, as the observed metric is the click ratio (CTR), we propose a mixture model of Generalized Linear Model (GLM) to model longitudinal data. Then, taking into account the resulting ad campaigns clusters, we develop a methodology to predict a probability of click. The proposed model describes each advertising campaign c as longitudinal data. The CTR is the target variable.

Each day is divided into H time slots. Each campaign c is observed J_c days and some slots could be missing. Let us consider Y_{cjh} the number of clicks for each campaign c at a specific time slot (j,h), for $j = 1,\dots,J_c$ and $h = 1,\dots,H$. We assume that each variable Y_{cjh} follows a distribution of the exponential family as introduced by [11], defined in Equation (1).

$$\forall c = 1,\dots,C, \forall j = 1,\dots,J_c, \forall h = 1,\dots,H$$
$$f_{Y_{cjh}}(y_{cjh},\theta_{cjh},\psi) = \exp\left(\frac{y_{cjh}\theta_{cjh} - b(\theta_{cjh})}{a_{cjh}(\psi)} + d(y_{cjh},\psi)\right) \tag{1}$$

where θ_{cjh} is the canonical parameter, ψ the dispersion parameter. b and d are specific functions depending on the exponential distribution chosen. We can define a_{cjh} as $a_{cjh}(\psi) = \frac{\psi}{\omega_{cjh}}$ with ω_{cjh} a weighted parameter for each observation.
Let us consider a binomial distribution for Y_{cjh} with paramaters n_{cjh} and $p_{hs(c,j)}$:

$$Y_{cjh} \sim \mathcal{B}(n_{cjh}, p_{hs(c,j)}) \tag{2}$$

where n_{cjh} is the number of observed impressions, $p_{hs(c,j)}$ the click probability of campaign c at time slot h and day of week $s(c,j)$, with $s(c,j) = 1,...,S$. We have a focus on the ratio $\frac{Y_{cjh}}{n_{cjh}}$ for the following.
In the case of the binomial, $\theta_{cjh} = log\left(\frac{p_{hs(c,j)}}{1-p_{hs(c,j)}}\right)$. We can define a, b and d functions as follows: $a_{cjh}(\psi) = \frac{1}{n_{cjh}}$, $b(\theta_{cjh}) = \log(1+\exp\theta_{cjh})$ and $d(y_{cjh},\psi_{cjh}) = \log\binom{n_{cjh}}{y_{cjh}}$.
We define the logit function $\eta = g(\mu) = \log\frac{\mu}{1-\mu}$ where μ is the expectation of ratio Y/n. Then, the function links the linear combination of the β parameters with the expectation $E(\frac{Y_{cjh}}{n_{cjh}})$:

$$\log\left(\frac{E(Y_{cjh}/n_{cjh})}{1-E(Y_{cjh}/n_{cjh})}\right) = \beta_0 + \beta_h^H + \beta_{s(c,j)}^S \tag{3}$$

In Equation (3), β_h^H and $\beta_{s(c,j)}^S$ coefficients are associated to the 2 categorical variables: time slot and day of the week. In the following, the model will be extended by other advertising context variables (see Section 3.4).

2.2 Mixture of binomial for ads clustering

The objective of our approach is to obtain a mixture model of binomial distributions. Considering that C campaigns come from K subpopulations, the mixture allows to model the heterogeneity of an overall population. We denote for each campaign c the density function $f_k(y_c;\phi_k), k = 1,...,K$ with the model parameters $(\phi_1,...,\phi_K)$. Campaign density function can also be written as following: $f(y_c) = \prod_{j=d_c}^{f_c}\prod_{h=1}^{H} f(y_{cjh})$, for all $k = 1,\dots,K$ where d_c and f_c respectively are the first and last day of diffusion observed for the campaign c. We assume that a campaign belongs to the same subpopulation throughout the time.
The considered mixture model is:

$$f(y_c;\phi,\lambda) = \sum_{k=1}^{K}\lambda_k f_k(y_c;\phi_k) \tag{4}$$

where $(\lambda_1,...,\lambda_K)$ are the mixing proportion with $\lambda_k \in (0,1)$ for all k and $\sum_{k=1}^{K}\lambda_k = 1$. The log-likelihood $\log L$ is written:

$$\log Ln(Y;\phi,\lambda) = \sum_{c=1}^{C}\log\left\{\sum_{k=1}^{K}\lambda_k f_k(y_c;\phi_k)\right\} \tag{5}$$

For parameter estimation, the mixture model defined in Equation (5) can be considered as an incomplete data structure model. We introduce the hidden variable Z_{kc} where $Z_{kc} = 1$ when campaign c belongs to cluster k and 0 otherwise. Using the hidden variables, the

log-likelihood for complete data is easier to manipulate for estimation:

$$\log Ln(Y, Z; \phi, \lambda) = \sum_{c=1}^{C} \left\{ \sum_{k=1}^{K} Z_{kc} \log \left(\lambda_k f(y_c; \phi_k) \right) \right\} \quad (6)$$

The most popular way to estimate model parameters is to solve iteratively likelihood equations in order to obtain maximum likelihood estimation for each parameter. When we do not have the analytic expression of the log likelihood, the most efficient algorithms to obtain parameters estimation are the Expectation-Maximization (EM) type algorithms introduced by [4].

E-Step For each iteration, the E-Step calculates the expectation of complete-data likelihood conditionally to observed data y and current parameters $\{\lambda_k,\ \phi_k\}_{k=1,\dots,K}$. We consider $Q(\phi|\phi^{(m)}) = E(\log Ln(Y, Z; \phi, \lambda)|Y = y, \phi^{(m)})$ at iteration m. As $E(Z_{kc}|Y_c, \phi^{(m)}) = P(Z_{kc} = 1|Y_c, \phi_k^{(m)})$, thanks to Bayes formula, we calculate :

$$\begin{aligned} \pi_{kc} &= P(Z_{kc} = 1|Y_c, \phi_k^{(m)}) && (7) \\ &= \frac{P(Y_c|Z_{kc} = 1, \phi^{(m)})P(Z_{kc} = 1)}{\sum_{l=1}^{K} P(Y_c|Z_{lc} = 1, \phi^{(m)})P(Z_{lc} = 1)} && (8) \\ &= \frac{f_{\phi_k^{(m)}}(y_c)\lambda_k}{\sum_{l=1}^{K} f_{\phi_l^{(m)}}(y_c)\lambda_l} && (9) \end{aligned}$$

The probability π_{kc} represents the probability that the campaign c belongs to the cluster k at iteration m. $\pi \in \mathbf{M}_{k\times n}$ is a matrix of probabilities where the sum of each column is equal to one.

M-Step The M Step consists in maximizing $Q(\phi|\phi^{(m)})$ in order to update the model parameters. As we model a mixture of binomial, there is no explicit solution for the β_k parameters. We use the iterative Fisher algorithm [14] to estimate β_k at each M Step :

$$\beta_k^{(m+1)} = \beta_k^{(m)} - \left(E\left[\frac{\partial^2 Q(\phi|\phi^{(m)})}{\partial\beta_k \partial\beta_{k'}} \right] \right)^{-1} \frac{\partial Q(\phi|\phi^{(m)})}{\partial\beta_k} \quad (10)$$

This algorithm is based on Newton-Raphson algorithm, in which the search direction of the new value $\left(-\frac{\partial^2 Q(\phi|\phi^{(m)})}{\partial\beta_k \partial\beta_{k'}} \right)$ is replaced by its expectation. We recognize the Fisher Information expression.
Using the mixture model of binomial defined in Equation (1) and (3), a Fisher algorithm iteration from Equation (10) leads to the following estimation of parameters $\beta_k^{(m+1)}$:

$$\begin{aligned} &\beta_k^{(m+1)} = \left(\sum_{c=1}^{C} \pi_{kc} M_c^t W^{-1}_{c\beta_k^{(m)}} M_c \right)^{-1} \times \\ &\sum_{c=1}^{C} \pi_{kc} M_c^t W^{-1}_{c\beta_k^{(m)}} \left[M_c \beta_k^{(m)} + \frac{\partial\eta_{kc}}{\partial\mu_k} \left(\frac{Y_c}{n_c} - \mu_k \right) \right] \end{aligned} \quad (11)$$

with M_c the design matrix of the campaign c, μ_k the ratio of click (CTR) expectation in cluster k. The diagonal matrices are defined : $W_{c\beta_k^{(m)}} = diag\left(\frac{1}{n_{cjh}} \frac{(1+\exp M_{cjh}\beta_k^{(m)})^2}{\exp M_{cjh}\beta_k^{(m)}} \right)_{cjh}$ and $\frac{\partial\eta_k}{\partial\mu_k} = diag\left(\frac{(1+\exp M_{cjh}\beta_k^{(m)})^2}{\exp M_{cjh}\beta_k^{(m)}} \right)_{cjh}$
For each step M, we repeat a few iterations of the Fisher scoring algorithm.

This GLM model-based clustering is implemented in the R package *binomialMix* (available on CRAN[2]). Note that the number of observations n_c for each campaign c do not need to be equal.

2.3 Predict CTR from clustering results

According to the problem statement described in the Section 1, the final goal is to predict the probability of click for each campaign c. The predictions are made in order to choose one campaign c as soon as there is an advertising position available. We consider five predictive approaches. Three are considered as baseline: two naive baseline predictions and a standard GLM (without mixture). The other two predictive models are based on GLM mixture estimations.

A) A vector of zeros Data is very imbalanced: 71% of the CTR value equals zero. The most naive baseline is to consider a CTR prediction always equal to 0 no matter what the context is.

B) Yesterday's CTR This second approach is an other naive way to model the prediction. We consider that for each observation Y_{cjh}, the observed CTR at exactly the same time slot but one day before is the predicted CTR for the current moment. We make the hypothesis that from one day to another, CTR values remain stable in a similar context.

C) Binomial predictive model We consider a classical Generalized Linear Model with a binomial distribution as described in equations (2) and (3). With this simple modeling, we analyze if there is a relevant linear combination of features able to predict a click probability for any given context for all the campaigns.

We consider these three models A), B) and C) as baselines.

D) Mixture of binomial The most intuitive methodology to implement from clustering results is to use the estimated β_k from each cluster. With these estimations, we can naturally obtain prediction for each cluster of campaigns.

E) Mixture of binomial + individual random effect for each cluster We now assume that the n_c observations ($n = \sum_{c=1}^{C} n_c$) from one campaign c are no longer independent. For each target CTR value y, we define a random effect ξ_c to model dependence of observations from a same campaign c. Lets consider η the logit function defined for Equation (3). The Generalized Linear Mixed Model (GLMM) for the C campaigns can be written

$$\eta_\xi = M\beta + U\xi \quad (12)$$

where η_ξ is defined from linked function $g : \eta_\xi = g(\mu_\xi)$ with $\mu_\xi = E(Y|\xi)$. $\beta \in \mathbf{R}^B$ is the vector of the B fixed effects. M is the design matrix associated. We denote $\xi \in \mathbf{R}^C$ the random effect vector of size C and U the design matrix. We suppose that ξ_c follows a Gaussian distribution $\mathcal{N}(0, \sigma^2\mathbf{I}_c)$. Conditionally to ξ, the model has the same properties as the GLM from Equation (2), (3).
In this approach, we run the GLMM model (see Equation (12)) for each cluster using the R package *lme4*. The model estimates fixed effects β as well as the random effect ξ, i.e. specific coefficients for each campaign present in the cluster in question. Once all the parameters are estimated, we can calculate predictions for each campaign and for each cluster.

[2] https://cran.r-project.org/web/packages/binomialMix/index.html

3 Experiments

3.1 Dataset

We consider a dataset from TabMo's[3] real traffic. The platform provides us with a very large volume of data as the incoming traffic hits a million bid requests per second. The data has first been preprocessed in order to obtain the expected format. We aggregate by campaign, time slot, day of week and ad slot features the observed number of clicks and impressions (number of times a given advertising is seen). Some campaigns last few days and others are displayed for months. The whole dataset contains 70123 rows and 12 columns. An extract of 4 rows randomly chosen is available in Table 1. We differentiate 2 types of variables:

1. The response variable CTR (in bold) is calculated as the ratio of the number of clicks (Y_{cjh}) and the number of impressions (n_{cjh}). The data is very imbalanced with around one click for 1000 impressions.

2. The other variables are the explainable features used for the modeling. Most of them are categorical as described in Table 2. *Time slot* contains 6 different labels (00h-7h,7h-12h,12h-14h,14h-18h,18h-00h) defined by domain experts and extracted from *timestamp* variable. The *ID* column is a distinct of all the 373 observed campaigns for the dataset that we want first to classify and then predict CTR.

Table 1. Extracted rows from Dataset which is composed of 70123 rows, 12 columns and 373 advertising campaign

ID	*Timestamp*	DayWeek	TimeSlot	OS	Support
622	*2018-11-20*	3	3	Android	Site
622	*2018-11-20*	3	4	Android	Site
377	*2019-01-26*	7	2	Android	App
101	*2018-12-02*	1	4	iOS	App

Ad Type	Ad Length	Ad Height	**Impressions**	**Clicks**	**CTR**
banner	320	480	**31**	**0**	**0**
banner	320	480	**57**	**1**	**0.017**
custom	320	480	**180**	**2**	**0.011**
banner	300	250	**64**	**0**	**0**

Table 2. Description of explainable features used for the model-based clustering

	Type	#Label	Description
1-Day of Week	Categorical	7	Monday to Sunday
2-Time Slot	Categorical	6	Slots of few hours
3-OS Type	Categorical	3	iOS, Android, Other
4-Support Type	Categorical	2	Application, Site
5-Advertising Type	Categorical	3	Example : Type 1
6-Advertising Length	Numerical		Pixels dimension
7-Advertising Height	Numerical		Pixels dimension

3.2 Evaluation metrics

Model choice metrics To select the right number of clusters in the mixture model, we use the BIC criterion [17]. The selected model is the one that minimizes its value. BIC is defined:

$$BIC = -2 \times (\log \hat{L}) + m \times \log(n) \tag{13}$$

where $(\log \hat{L})$ is the maximized value of the incomplete log-likelihood defined in Equation (5). m is the global number of parameters for the model and n the total number of observations in the dataset.
We can also use the Integrated Complete Likelihood (ICL) criterion [1] which is an adaptation of the BIC dedicated to clustering.

Clustering robustness metrics In order to compare clustering results, we use Adjusted Rand Index (ARI) introduced by [7]. It is based on Rand Index (RI) [15] which is a measure of similarity between two partitions and calculates the percentage of pairs in agreement. The RI and ARI values are between 0 and 1. A Rand Index (or Adjusted Rand Index) equal to 1 corresponds to two identical clustering partition. The Adjusted Rand Index is a corrected version of the Rand Index. The calculation of this index is presented in Equation (14) with notation from the contingency table described in Table 3. In this table, we consider two partitions P and Q with respectively k and j clusters.

$$ARI = \frac{\sum_{l,k}\binom{n_{jk}}{2} - \left[\sum_l \binom{n_{l.}}{2}\sum_k \binom{n_{.k}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_l \binom{n_{l.}}{2} + \sum_k \binom{n_{.k}}{2}\right] - \left[\sum_l \binom{n_{l.}}{2}\sum_k \binom{n_{.k}}{2}\right]/\binom{n}{2}} \tag{14}$$

Table 3. Contingency table for two clustering partitions P and Q

	Partition 1				
Partition 2	p_1	p_2	...	p_k	Sums
q_1	n_{11}	n_{12}	...	n_{1k}	$n_{1.}$
q_2	n_{21}	n_{22}	...	n_{2k}	$n_{2.}$
...	...	...	$\ddots$	...	$\vdots$
q_l	n_{l1}	n_{l2}	...	n_{lk}	$n_{l.}$
Sums	$n_{.1}$	$n_{.2}$	...	$n_{.k}$	$\mathbf{n} = n_{..}$

Predictive accuracy metric In order to evaluate the predictive accuracy, we consider the logloss [13]. The logloss is used to calculate the difference between a prediction and its associated target variable (which is between 0 and 1). For example, if a model predicts a probability $\hat{p} = 0.004$ and that true observation is 1, the logloss will be very bad. Logloss increases as the predicted probability diverges from the actual label. In our case, each observation is an aggregation. We study the number of times an ad is clicked (y_{cjh}) among the number of times the ad is displayed (n_{cjh}) for each campaign c at a specific time j, h (see Equation (2)). We define the number of "no click" : $n_{cjh} - y_{cjh}$. The logloss ($LogLoss_{cjh}$) is calculated for each observation of the dataset (Equation (15)).

$$LogLoss_{cjh} = -(y_{cjh}\log\hat{p} + (n_{cjh} - y_{cjh})\log(1-\hat{p})) \tag{15}$$

The resulting value that we want to analyze is the **mean logloss** :

$$LogLoss = \frac{\sum_c \sum_j \sum_h LogLoss_{cjh}}{\sum_c \sum_j \sum_h n_{cjh}} \tag{16}$$

where the numerator is the sum of logloss for each aggregated observations and the denominator is the total number of impressions.

[3] https://hawk.tabmo.io/

3.3 A short simulation study

Before evaluation on real data with significant size, we carry out a two-step simulation study: in the first step, we try to find the right partition when we know the model. In the second step, the objective is to find both the right model and the right partition.

We first assess the ability of our approach to find the right partition when the Generalized Linear Model is known. We simulate ratio of clicks for C (here, $C = 373$) advertising campaigns, uniformly distributed in $K = 2$ to 5 clusters. For each cluster, the CTR is simulated according to a binomial distribution with only 2 explanatory variables: day and time slot. Expectation of the rate of clicks is selected in 3 different intervals ($[0.2, 0.5]$, $[0.1, 0.2]$, and $[0.01, 0.1]$) so that we can estimate the impact of a low CTR in the modeling. 10 simulations were carried out in each situation. The results are presented in Figure 1. The number of clusters is correctly estimated for 2 and 3 simulated clusters, regardless of the click rate expectation. From 4 simulated clusters, the number of clusters is not always correctly estimated, especially since the expectation of the CTR is low. This is an expected behavior of the model since there are fewer campaigns involved in parameter estimation in each class. Looking at the

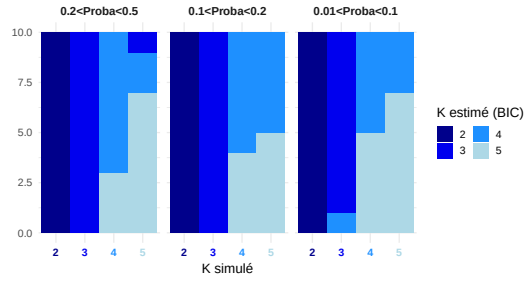

Figure 1. Comparison of the number of clusters simulated and estimated by BIC when the model is known

Adjusted Rand Index in Figure 2, conclusions are mostly identical. Indeed, up to 4 simulated clusters, for a click rate expectation greater than 0.1, the estimated partitions are very close to the simulated partitions. The partition estimation quality deteriorates for a click rate expectation of less than 0.1, even for a small number of clusters. According to these first simulations, a data set of 373 campaigns allows to correctly identify up to 3 to 4 clusters, for a CTR expectation higher than 0.1 in the case of a binomial model defined by 11 free parameters.

We are now evaluating the ability of our approach to find the right

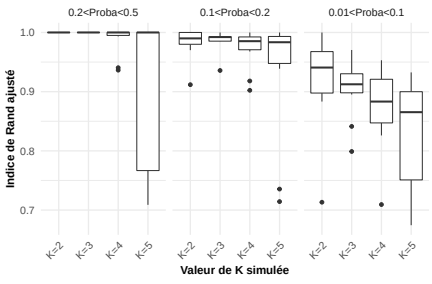

Figure 2. Adjusted Rand Index boxplot when the model is known

partition and model simultaneously. We simulated CTR for 400 advertising campaigns, uniformly distributed in K=2 to 5 clusters. The click rate is simulated according to a binomial distribution with different parameters: 2 explanatory variables (day and time slot), a single day feature, a single time slot feature, the intercept only. The expectation of the click rate is set in the interval $[0.2, 0.5]$. For each of the 8 simulations performed in each case, the model and partition chosen are the ones minimizing the BIC criterion. The results for the case K=4 and 2 explanatory variables model are presented in Table 4. The correct partition and model are found in 7 out of 8 cases, with just an error on the partition for the last case. The results leads us to the same conclusion in the other cases.

Table 4. Example of simulation where the number of clusters is equal to 4 and features used are day of week and time slot. The correct model and right number of clusters is retrieved in 7 out of 8 simulations.

	2	3	4	5
Day feature	0	0	0	0
Time Slot feature	0	0	0	0
Intercept only	0	0	0	0
Day feature + Time Slot feature	0	0	7	1

3.4 Results on Real Data

First results on real data concern mixture of binomial for advertising campaigns. Based on Equation (3) and features described in Table 2, the model (7 features, 19 free parameters) is defined in Equation (17) with β_{os}^{OS} corresponding to categorical feature OS type, β_{as}^{AS} to Support Type, β_{ad}^{AD} to Advertising Type (see Table 2). The advertising size (length and width) is measured by two numerical variables x_l and x_w.

$$\log\left(\frac{E(Y_{cjh}/n_{cjh})}{1-E(Y_{cjh}/n_{cjh})}\right) = \beta_0 + \beta_h^H + \beta_{s(c,j)}^S + \beta_{os}^{OS} + \beta_{as}^{AS} + \beta_{ad}^{AD} + \alpha_l x_l + \alpha_w x_w \quad (17)$$

For sake of clarity, we keep the index cjh which should be augmented by os, as, ad.

Optimal number of clusters The number of clusters varies from $K = 2$ to $K = 6$. We evaluate the number of clusters with the BIC criterion (Equation 13). Based on Figure 3, the optimal number of clusters is $K = 5$. Same evaluation is done with ICL and gives the same results with a minimum value for $K = 5$.

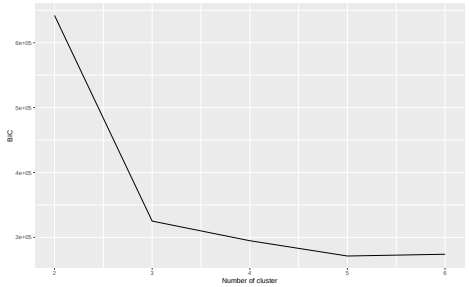

Figure 3. Evaluation of BIC for $K = 2$ to $K = 6$

Inferred profiles The resulting mixture model is composed of 5 groups divided as following: clusters 1, 2, 3, 4 and 5 contain respectively 39, 217, 29, 37 and 73 campaigns.
We analyze the inferred profiles for each cluster in Figure 4. As there are 7 *day of week* levels and 6 *time slot* levels, the x-axis represents the 42 combinations from those temporal features. The scale of the y-axis is the mean CTR estimated (in percentage).
Each figure corresponds to a cluster. From left to right on the top line, cluster 1, cluster 2 and cluster 3 are respectively displayed. On the bottom line, still from left to right, are displayed cluster 4 and cluster 5.
Figure 4 displays the mean estimated profiles for the 18 possible combinations of features levels, for banner width set to 320 and banner length to 480. The highlighted profile for cluster corresponds to the combination: Android, Application and Advertising Type 3. This is the configuration we want to compare. The scale of the y-axis is

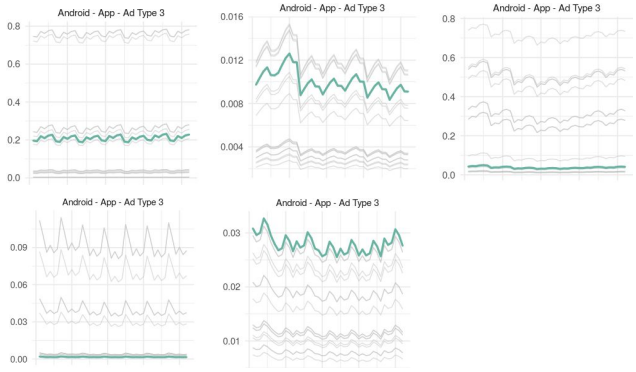

Figure 4. Estimation of inferred profiles for each cluster when Type of OS is Android, Type of support is Application, Ad Type is of Type 3 and Ad size is 320x480

different from one cluster to another. For cluster 1, the average CTR is around 0.2 while for the other clusters, the average CTR is well below 0.1. For any given combined levels, the inferred profiles are very different. We conclude that the clustering model groups advertising campaign with similar profiles and distinguishes specific types of campaigns from one group to another one. We analyze more in details two clusters. In Figure 5, eight different profiles are displayed. Dash line represents profiles when Support Type level feature is *Application*. Solid line is used for *Site* level. Red shaded lines correspond to Android profiles and blue ones to iOS.

- Cluster 1 groups campaigns with high CTR, especially for Site support and advertising of Type 3.
- Cluster 5 is composed of campaigns mainly affected by the App or Site feature, regardless of the type of advertising and the type of OS.

Prediction accuracy We evaluate the predictive performance of the models described in Section 2. Models (A) and (B) are naive modeling whose learning is respectively done from a vector of zeros or from information of the previous day. Model (C) is a generalized linear model with a logit link function with the features described in Table 2. Models (D) and (E) result from the mixture model: in model (D), we calculate the predictions based on the estimated β coefficients for each cluster. For model (E), for each cluster, we run a GLMM with a random "campaign" effect. To compare accuracy

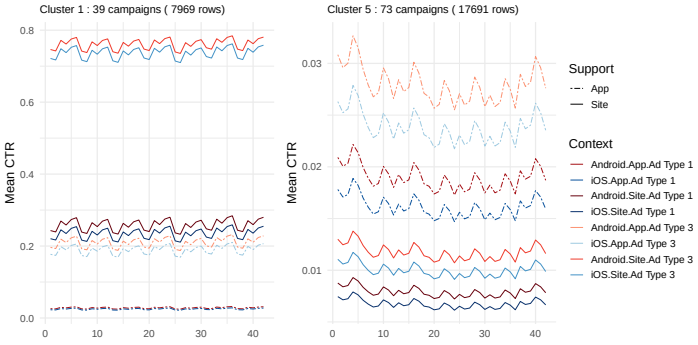

Figure 5. Inferred profiles for clusters 1 and 5. Cluster 1 groups campaign with high CTR, especially for Site support and advertising of Type 3. Cluster 5 is composed of campaigns mainly affected by the App or Site feature, regardless of the type of advertising and the type of OS.

of the models, we calculate the mean logloss (Equation 16) in two different cases:

1. **Test 1:** the dataset is extracted from one day randomly chosen (15-04-19). For models A and B, there is no learning set since we either use a vector of zeros or the vector of CTR observed the day before. For the three other models, the learning dataset is extracted from March 14th to April 14th, 2019.

   Mean Logloss for baseline Models (A) and (B) are 0.52 and 0.11. The binomial model (C) gives a mean logloss of 0.10. Models (D) and (E) resulting from the clustering step have a mean logloss respectively equal to 0.0812 and 0.0799. The addition of a random effect for each campaign ID seems to be relevant since the model (E) mean logloss outperforms other modeling.

2. **Test 2:** As the first test on one randomly chosen day provides good results, we repeat the same test procedure as before but on more days. We make the test and learning timestamp window evolve by shifting them from one day to the next one. For each new test/learning set, we run the test procedure. It allows to obtain a more global mean logloss since we learn and test on more distinct datasets. For this experiment, two periods of the year are studied: November/December and March/April. These two moments of year are very different. In November and December, activity on the bidding platform is very dense due to the end-of-year holidays, which generate a lot of advertising to display. March and April period is much calmer in terms of traffic observed on the platform. First, we shift from the first learning (01/11 - 30/11) and test set (01/12) to the last learning (30/11 - 30/12) and test set (31/12). Second, we do the learning on March/April. We shift from the first learning (01/03 - 31/03) and test set (01/04) to the last learning (30/03 - 29/04) and test set (30/04). The predictive procedure is done 30 times for both period. We analyze the resulting predictions.

For both periods, Model (A) is widely outperformed by the 4 others. The GLM mixtures outperforms the others: the mean logloss is the lowest with models (D) and (E). The clustering step with the mixture model seems to be relevant for the predicting step. Adding a random effect for each campaign in Model (E) provides a better logloss compared to the prediction with model-based clustering only. Even if it seems to be a small improvement in terms of logloss evaluation, it can lead to a significant increase for the company. In Figure 6 , we

Table 5. Mean logloss value for five models described in Section 2.3. In the first column, we calculate the mean Logloss for 30 days in December 2018. In the last column, the mean logloss is calculated for 30 days in April 2019

	Mean Logloss (December)	Mean Logloss (April)
Model (A)	0.2041	0.3468
Model (B)	0.0572	0.0948
Model (C)	0.0465	0.0711
Model (D)	0.0413	0.0598
Model (E)	**0.0405**	**0.0592**

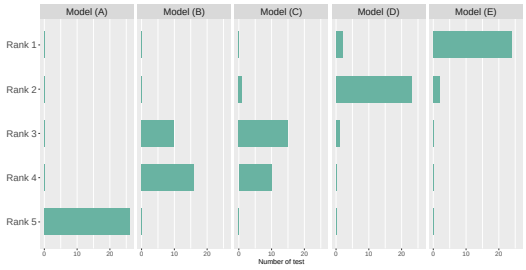

Figure 6. Ranking of the five models (see Section 2.3) obtained for each of the 30 tests performed in December 2018. Rank 1 corresponds to a model whose logloss is minimized compared to other models. Rank 5 corresponds to a model that has the worst logloss for a given test.

analyze more in details the mean logloss obtained for each models and each test day of December 2019. Rank 1 corresponds to a model whose logloss is minimized compared to other models. Rank 5 corresponds to a model that has the worst logloss for a given test. Model (A) always provides the worst mean logloss. Model (D) and (E) resulting from the clustering step almost always outperform the three other models. We obtain the same conclusions as before for Table 5: a preliminary model-based clustering step improves the prediction accuracy according to the mean logloss.

4 Conclusion and Perspectives

In this paper, we proposed a two-step methodology for the prediction of CTR in an industrial context.
First objective was to obtain a mixture model for binomial and longitudinal data. In the second step, several predictive models were in competition. Three were considered as baselines and the two others used estimated coefficients from each cluster to predict a probability of click. Using a preliminary clustering step before prediction improved the performance of prediction with a relevant logloss decrease. For future work, we want to study the optimal history window necessary for the learning step. We also want to expand the model by adding new contextual features such as the IAB category[4] for each application/site. It could be useful for the predictive task to consider second order interactions between features.
For further experiments, the model will be implemented in the large scale auction system. The objective is to evaluate its performance by A/B testing feedback in production.

[4] https://www.iab.com/wp-content/uploads/2016/03/OpenRTB-API-Specification-Version-2-5-FINAL.pdf

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments, which helped improve this paper.

REFERENCES

[1] Christophe Biernacki, Gilles Celeux, and Gérard Govaert, 'Assessing a mixture model for clustering with the integrated completed likelihood', *IEEE transactions on pattern analysis and machine intelligence*, **22**(7), 719–725, (2000).
[2] Olivier Chapelle, Eren Manavoglu, and Romer Rosales, 'Simple and scalable response prediction for display advertising', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **5**(4), 61, (2015).
[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al., 'Wide & deep learning for recommender systems', in *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10. ACM, (2016).
[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin, 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–22, (1977).
[5] C Fraley, AE Raftery, L Scrucca, TB Murphy, M Fop, and ML Scrucca. Gaussian mixture modelling for model-based clustering, classification, an density estimation, 2018.
[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He, 'Deepfm: a factorization-machine based neural network for ctr prediction', *arXiv preprint arXiv:1703.04247*, (2017).
[7] Lawrence Hubert and Phipps Arabie, 'Comparing partitions', *Journal of classification*, **2**(1), 193–218, (1985).
[8] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin, 'Field-aware factorization machines for ctr prediction', in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 43–50. ACM, (2016).
[9] Gouthami Kondakindi, Satakshi Rana, Aswin Rajkumar, Sai Kaushik Ponnekanti, and Vinit Parakh, 'A logistic regression approach to ad click prediction', *Mach Learn Class Project*, (2014).
[10] Friedrich Leisch, 'Flexmix: A general framework for finite mixture models and latent glass regression in r', (2004).
[11] P McCullagh and John A Nelder, *Generalized Linear Models*, volume 37, CRC Press, 1989.
[12] Geoffrey McLachlan and David Peel, *Finite mixture models*, John Wiley & Sons, 2004.
[13] Kevin P Murphy, *Machine learning: a probabilistic perspective*, MIT press, 2012.
[14] John Ashworth Nelder and Robert WM Wedderburn, 'Generalized linear models', *Journal of the Royal Statistical Society: Series A (General)*, **135**(3), 370–384, (1972).
[15] William M Rand, 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical association*, **66**(336), 846–850, (1971).
[16] Steffen Rendle, 'Factorization machines', in *2010 IEEE International Conference on Data Mining*, pp. 995–1000. IEEE, (2010).
[17] Gideon Schwarz et al., 'Estimating the dimension of a model', *The annals of statistics*, **6**(2), 461–464, (1978).
[18] Mixmod Team. Mixmod statistical documentation, 2008.
[19] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang, 'Deep & cross network for ad click predictions', in *Proceedings of the ADKDD'17*, p. 12. ACM, (2017).
[20] Shuai Yuan, Jun Wang, and Xiaoxue Zhao, 'Real-time bidding for online advertising: measurement and analysis', in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, p. 3. ACM, (2013).
[21] Robbin Lee Zeff and Bradley Aronson, *Advertising on the Internet*, John Wiley & Sons, Inc., 1999.